

Trabalho Prático 2 Ant Colony Optimization

Data de entrega: 15/10/2020

1 Introdução

O objetivo desse trabalho é desenvolver conceitos chaves para construções de soluções para problemas usando *Ant Colony Optimization* (ACO), envolvendo o entedimento e a implementação dos componentes básicos de um arcabouço de ACO, bem como a análise de sensibilidade dos seus parâmetros (como eles afetam o resultado final, a natureza de convergência, etc) e procedimentos para avaliação das soluções alcançadas.

Para esse trabalho, vocês devem elaborar soluções para o problema conhecido como *longest path problem*¹, que é um problema NP-hard. Dado um grafo $G(V, E)$, uma função $w : E \rightarrow \mathbb{R}$ que atribui pesos a cada aresta e dois vértices $u, v \in V$, denotaremos como \mathcal{P} o conjunto de caminhos *simples*² partindo de u e chegando em v . O problema consiste então em encontrar $P^* = \{e_1^*, e_2^*, \dots, e_k^*\}$ tal que

$$P^* = \arg \max_{P \in \mathcal{P}} \sum_{e_i \in P} w(e_i) \quad (1)$$

Ou seja, queremos encontrar o caminho simples de u a v que maximize o peso total do caminho.

Decisões de Implementação:

- Como representar uma solução;
- Como calcular a fitness de uma solução;
- Como calcular a probabilidade de transição entre vértices;
- Como lidar com soluções inválidas.

2 Bases de Dados

Nesse trabalho, vocês deverão utilizar os três grafos fornecidos como entrada, conforme detalhado na Tabela 1. Todos os grafos são *direcionados* e ponderados, com pesos inteiros entre 1 e 10. Os vértices estão numerados de 1 à N . Para os dois primeiros grafos, sabemos a solução ótima. Vocês devem usar esses valores como referência para avaliar a solução proposta para o trabalho.

Tabela 1: Bases de Dados			
Base	Vértices	Arestas	Ótimo
1	20	190	168
2	100	8.020	990
3	1.000	499.500	-

Os arquivos de entrada seguem o seguinte formato: “id_no1 id_no2 peso”, ou seja, uma entrada “1 34 5” codifica uma aresta do nó 1 para o nó 34 com peso 5.

¹https://en.wikipedia.org/wiki/Longest_path_problem

²Um caminho simples é aquele que não passa pelo mesmo vértice mais de uma vez.

3 Metodologia Experimental

O ACO deve ser testado nas bases de dados descritas na Tabela 1. A avaliação experimental deve ser realizada para cada base de dados e inclui a análise de sensibilidade dos principais parâmetros do ACO (e de qualquer decisão de implementação não-usual) e a discussão dos resultados obtidos. Abaixo, temos uma lista não-exaustiva de parâmetros/decisões de implementação a se avaliar:

- Número de formigas;
- Número de iterações;
- Taxa de evaporação;
- Seu algoritmo gera soluções inválidas? Se sim, qual o impacto das possíveis maneiras de lidar com esse problema?

Lembrem-se que o ACO é um algoritmo *estocástico*. Logo, é necessário realizar diversas repetições de uma mesma configuração para podermos concluir qualquer coisa. O desvio-padrão é um bom indicador do número necessário de repetições e intervalos de confiança devem ser utilizados sempre que possível.

4 O Que Deve Ser Entregue

- Código-fonte;
- Documentação do trabalho:
 - Introdução;
 - Implementação: descrição sobre implementação do algoritmo, incluindo detalhes de representação, fitness, etc.
 - Experimentos: análise dos parâmetros/decisões no resultado do ACO e discussões de resultados;
 - Conclusões;
 - Bibliografia.

A entrega DEVE ser feita pelo Moodle na forma de um único arquivo zipado, contendo o código e a documentação do trabalho.

Considerações Finais

- Não é permitido o uso de bibliotecas de colônias de formigas para desenvolver o trabalho. O objetivo é que você entenda como o algoritmo funciona, e não como chamar as funções. Em caso de dúvidas, por favor entre em contato.
- Use seu bom senso e não copie o trabalho do colega. Uso um mecanismo de detecção de plágio. Trabalhos copiados serão zerados.
- Depois da entrega do trabalho, faremos uma competição virtual para avaliar as diversas decisões de implementação do algoritmo e como a otimização dos parâmetros podem levar ao sucesso ou fracasso do algoritmo.